epcc

Parallelism Inherent in the Wavefront Algorithm

Gavin J. Pringle

The Benchmark code

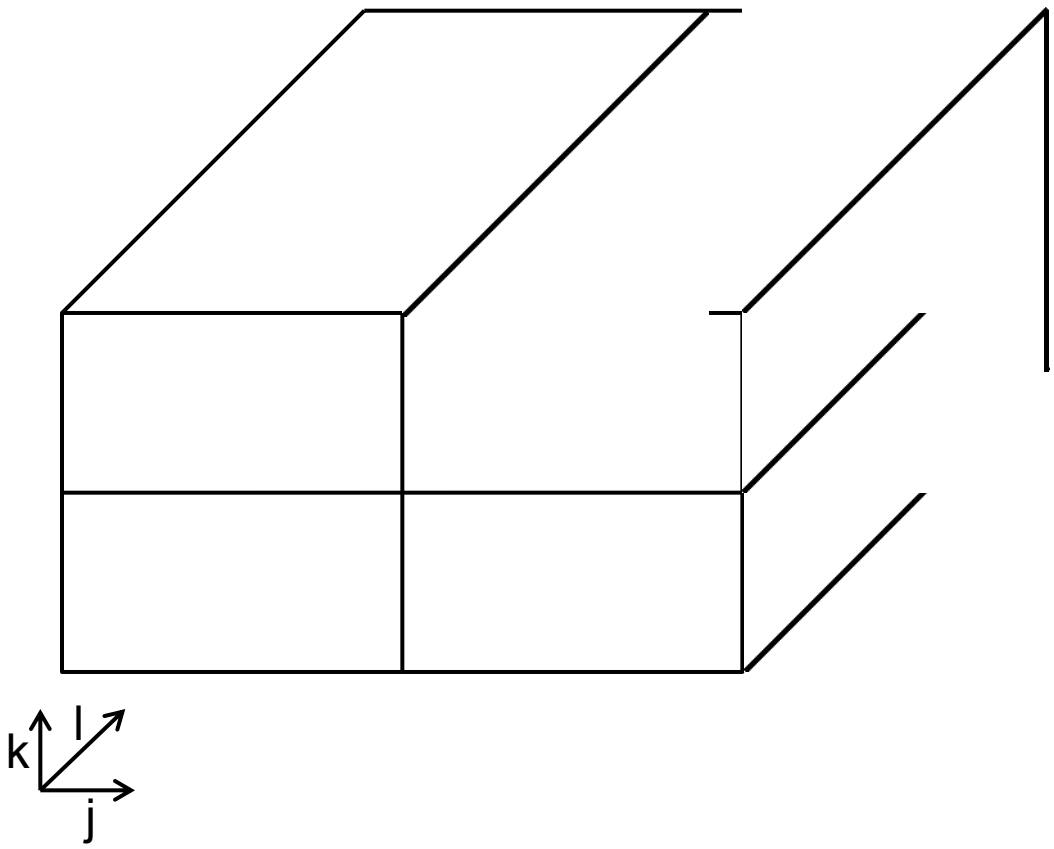
- Particle transport code using wavefront algorithm
  - Primarily used for benchmarking
- Coded in Fortran 90 and MPI
  - Scales to thousands of cores for large problems
- Over 90% of time in one kernel at the heart of the computation

Serial Algorithm Outline

Outer iteration
  Loop over energy groups
    Inner iteration
      Loop over sweeps
        Loop over cells in z direction
          Loop over cells in y direction
            Loop over cells in x direction
              Loop over angles (only independent loop!)
                work (90% of time spent here)
              End loop over angles
            End loop over cells in x direction
          End loop over cells in y direction
          …

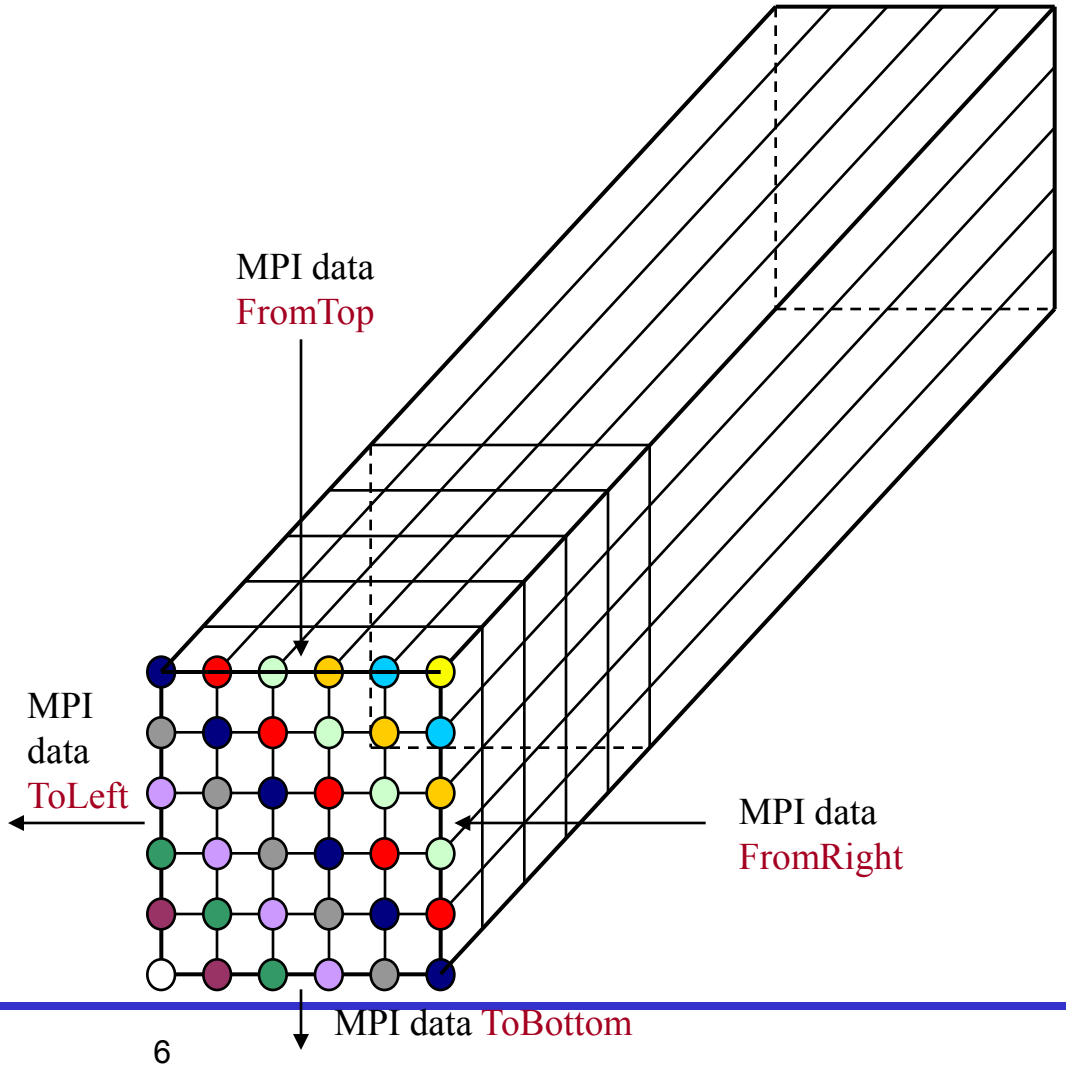
Close up of parallelised loops over cells

```
…
Loop over cells in z direction
    Possible MPI_Recv communications
    Loop over cells in y direction
        Loop over cells in x direction
            Loop over angles  (number of angles too small for MPI)
                work
            End loop over angles
        End loop over cells in x direction
    End loop over cells in y direction
    Possible MPI_Ssend communcations
End loop over cells in z direction
…
```



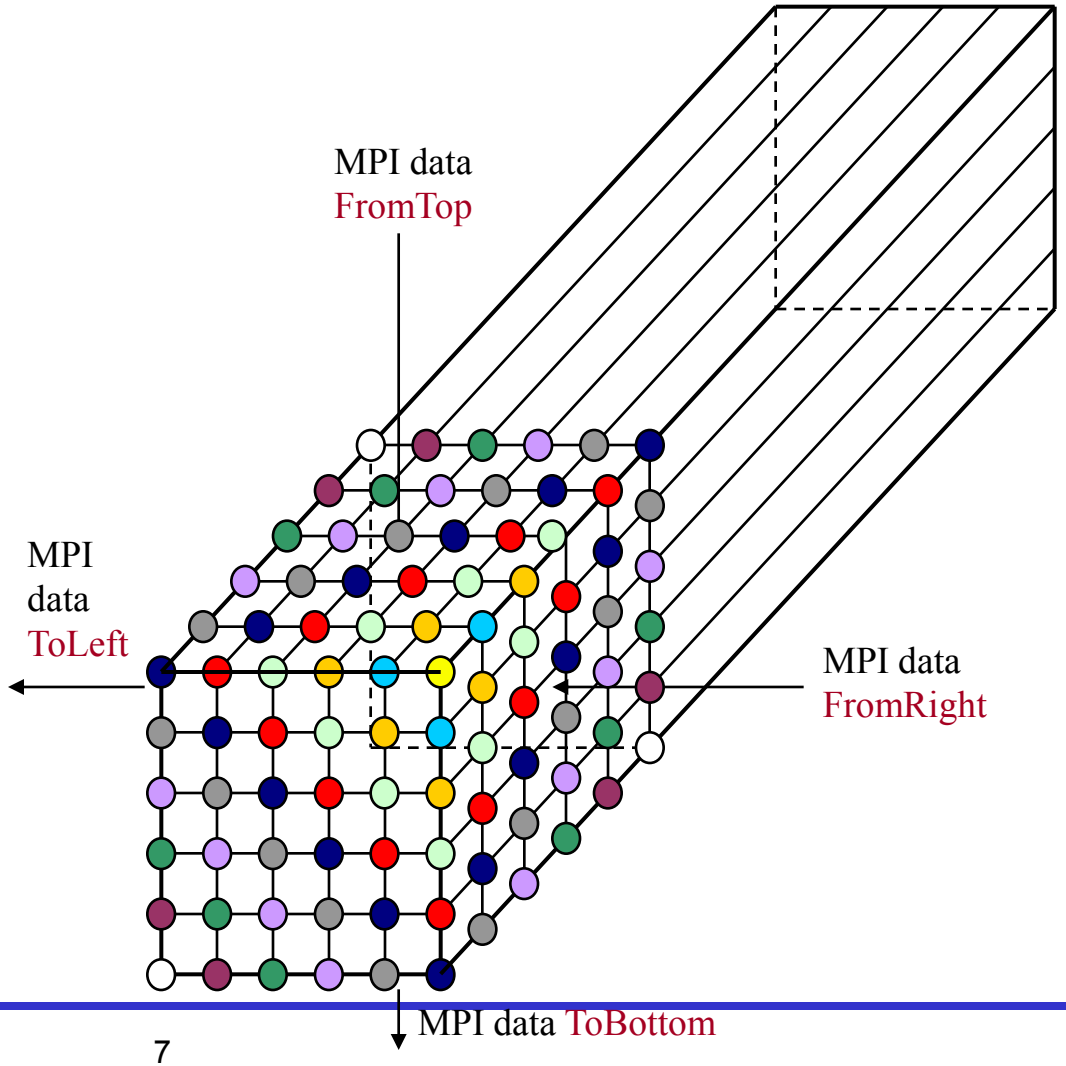
- MPI 2D decomposition is 2D decomposition of front x-y face.
- Figure shows 4 MPI tasks

Diagram of dependicies



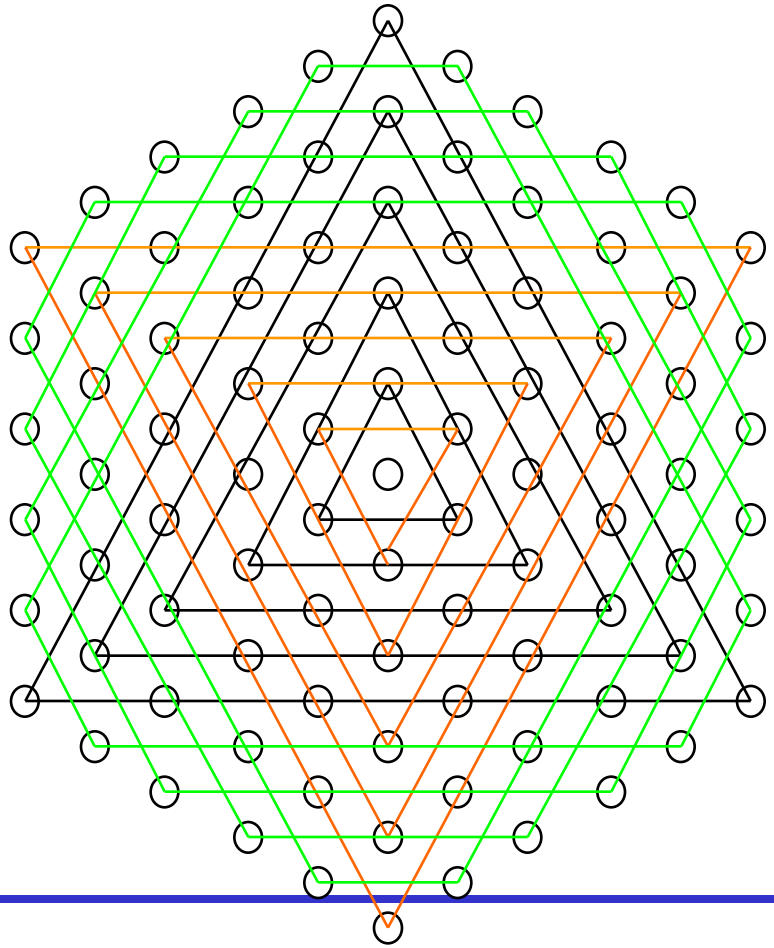
- This diagram shows **the domain of one MPI task**
- A cell cannot be processed until all cells “upstream” have been processed.

Sweep order: 3D diagonal slices



Cells of the same colour are independent and may be processed in parallel once preceding slices are complete.

Slice shapes (6x6x6)

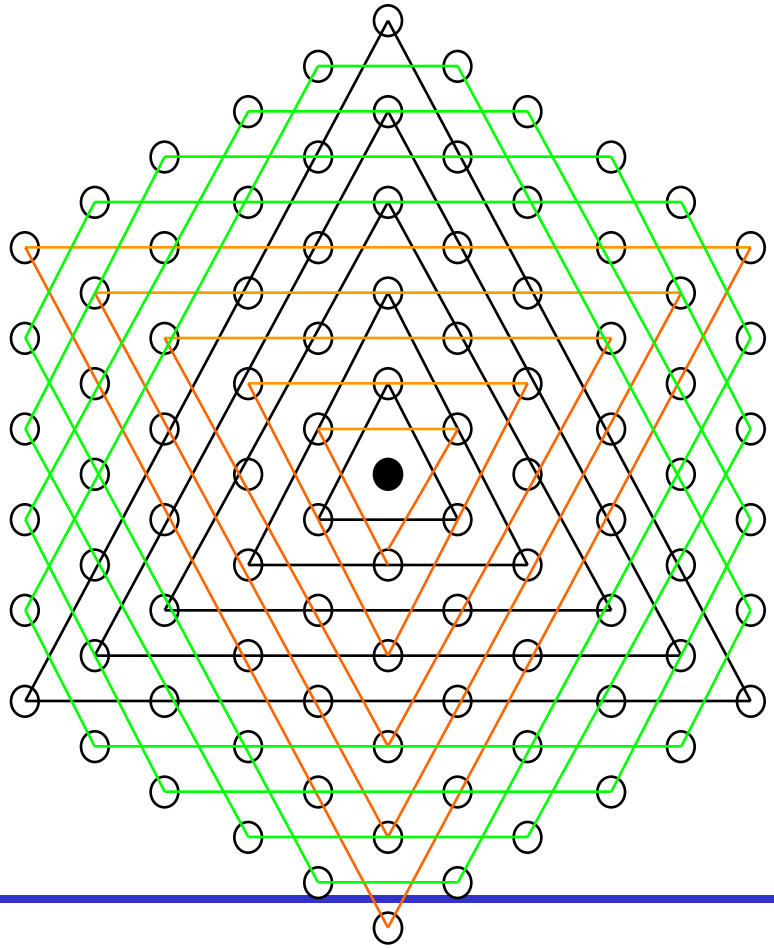


Increasing triangles

Then transforming Hexagons

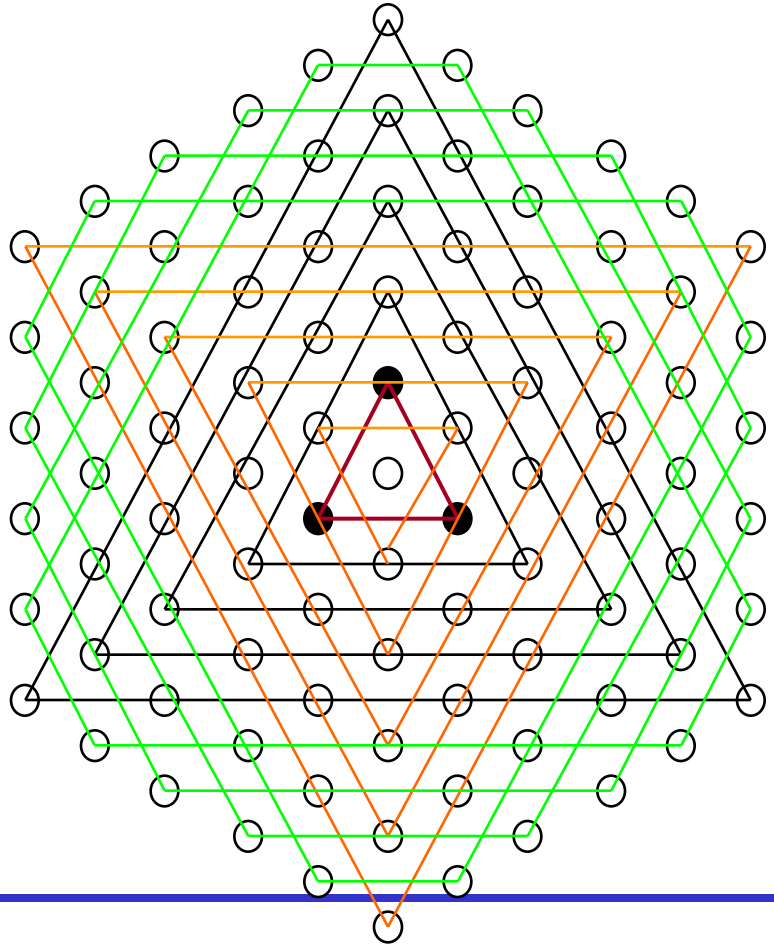
Then decreasing (flipped) triangles

Slice 1



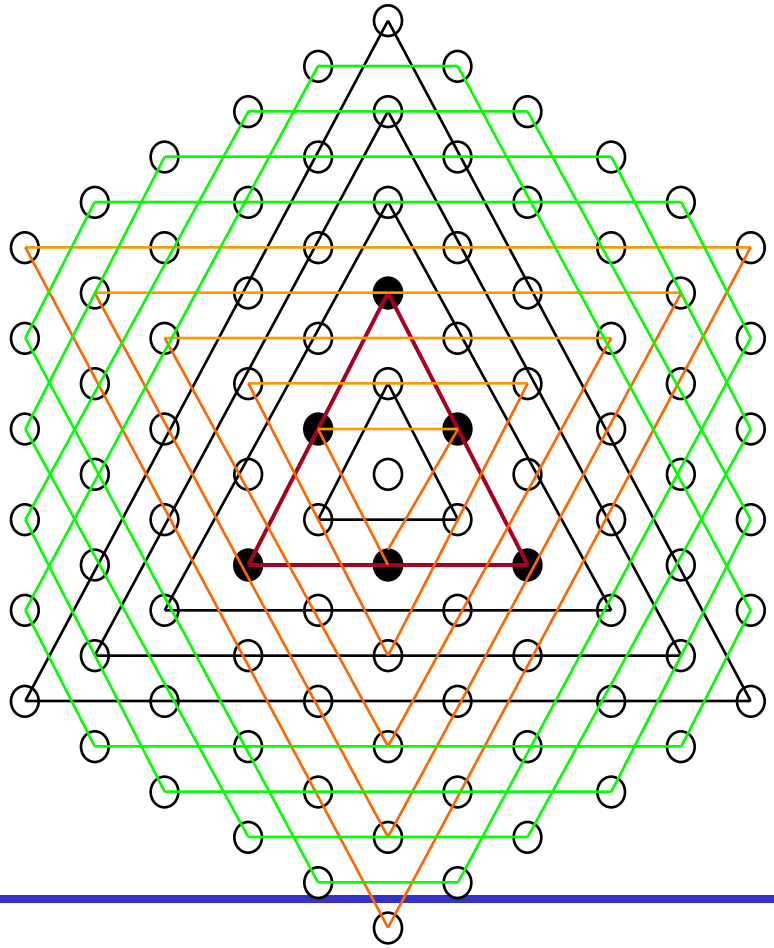
Cell nearest the viewer

Slice 2

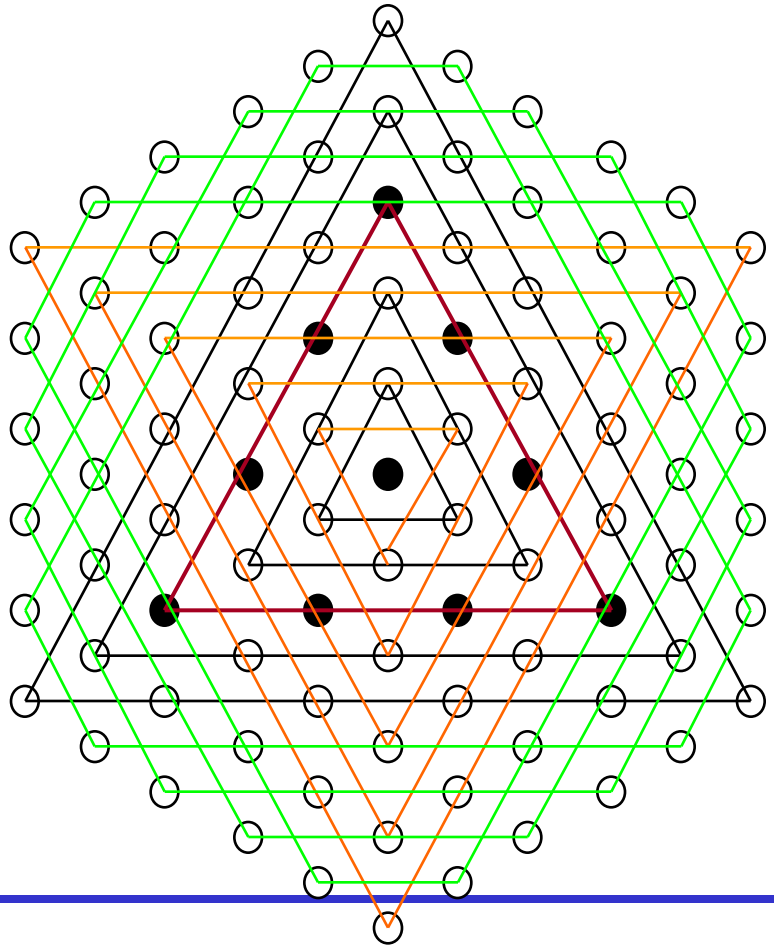


Moving down away from viewer

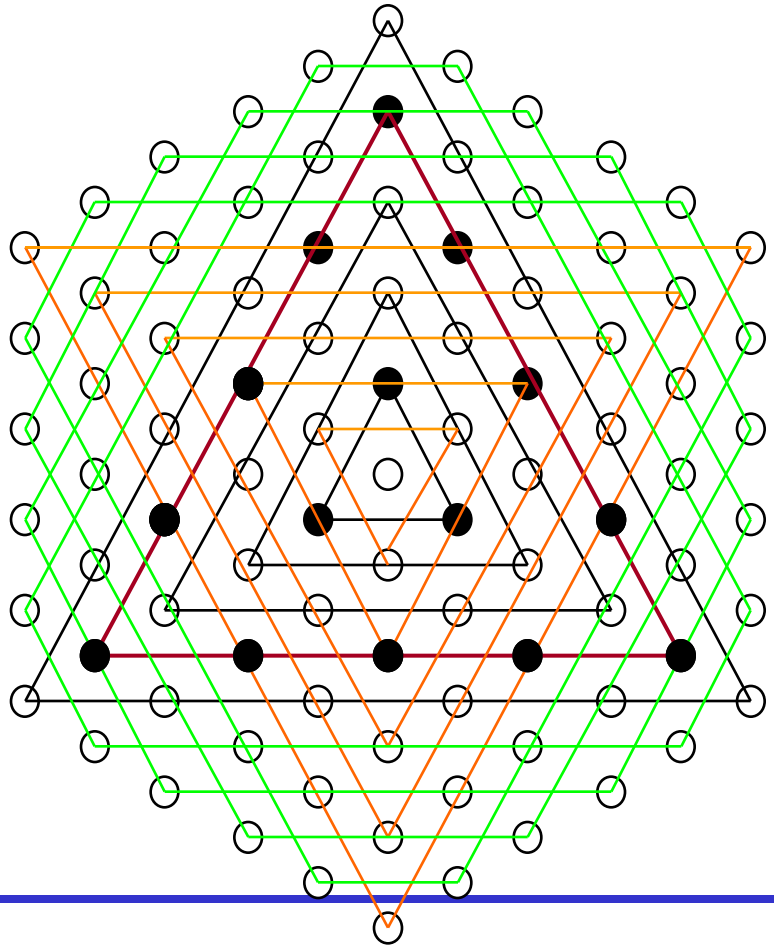
Slice 3



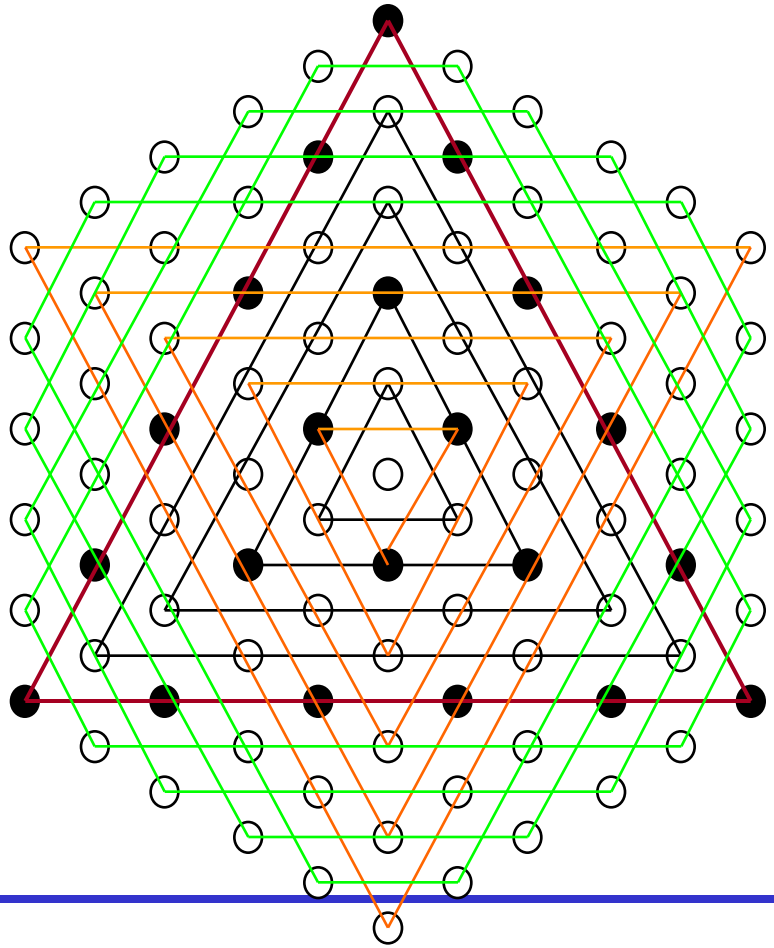
Slice 4



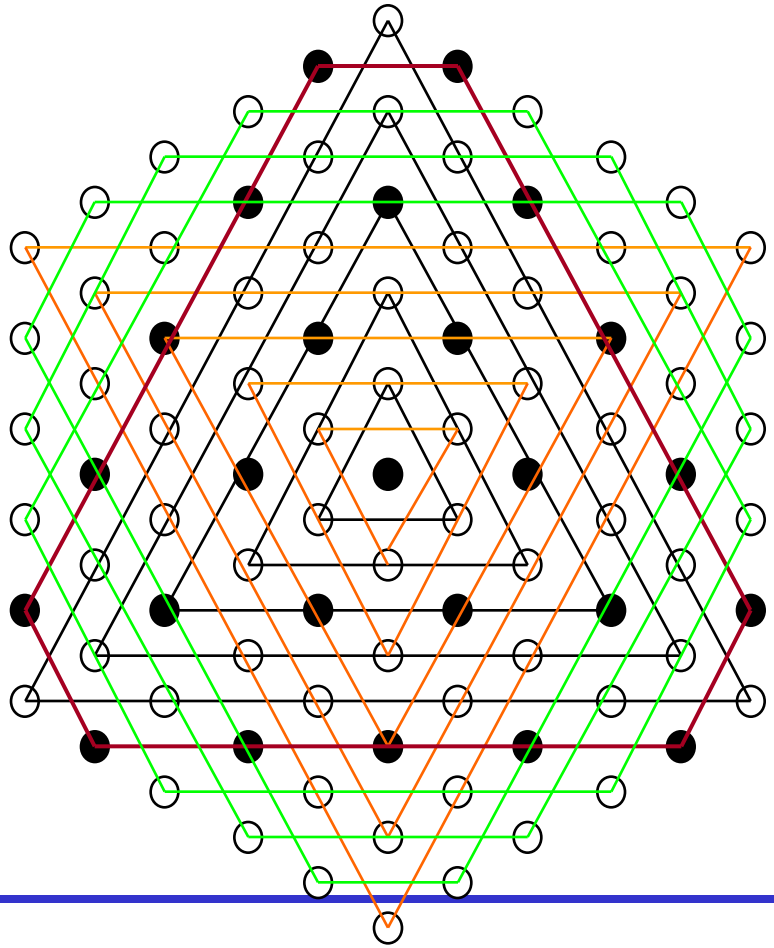
Slice 5



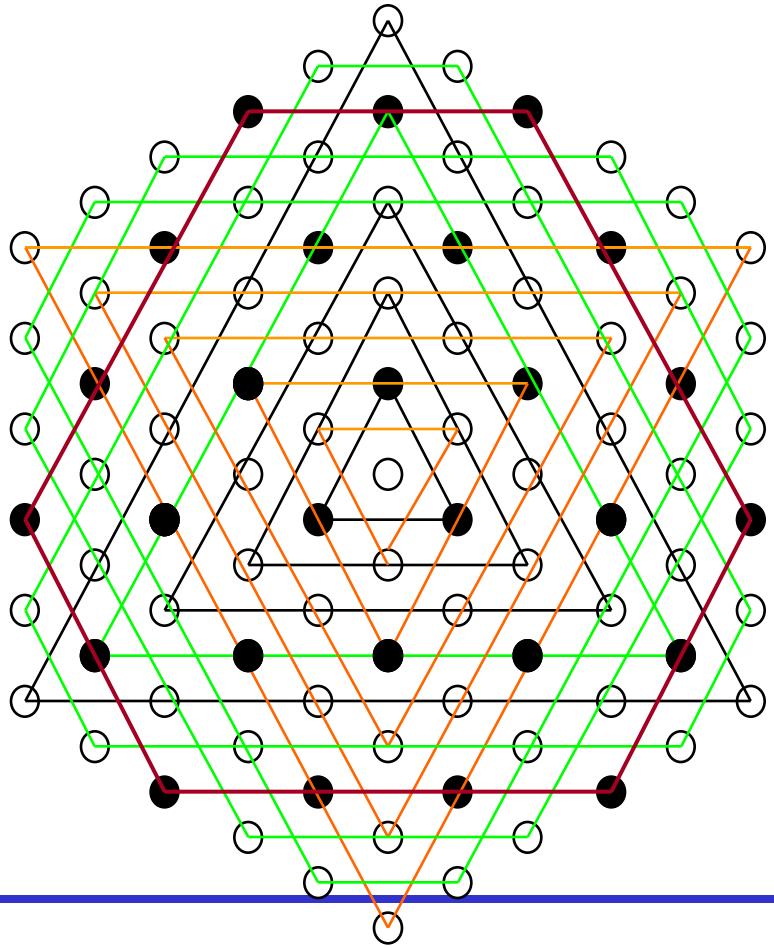
Slice 6



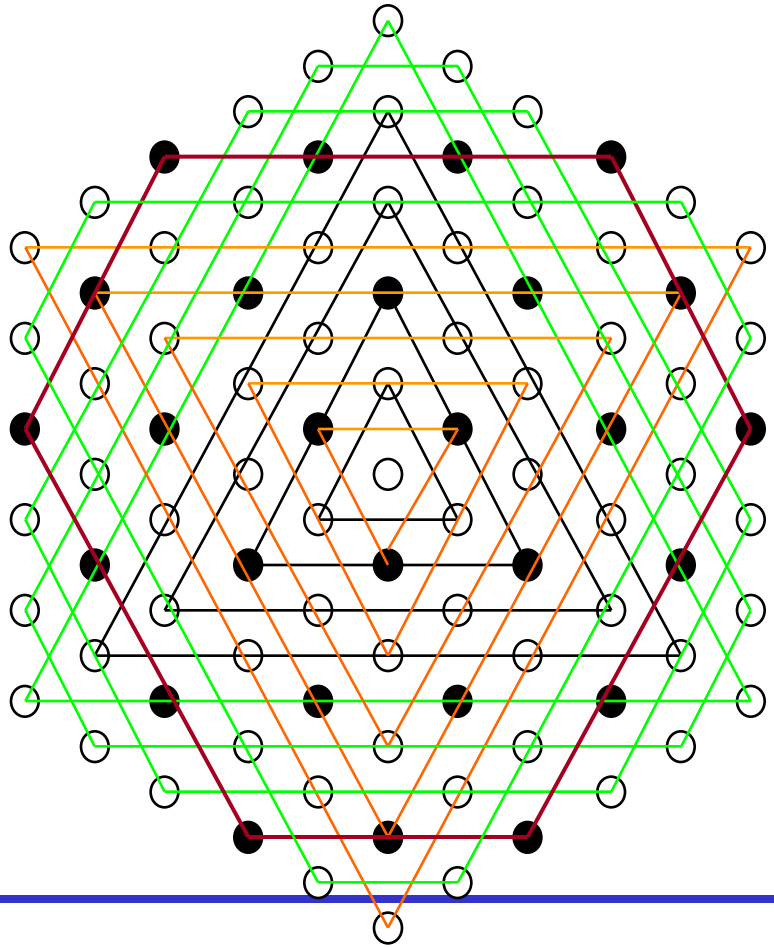
Slice 7



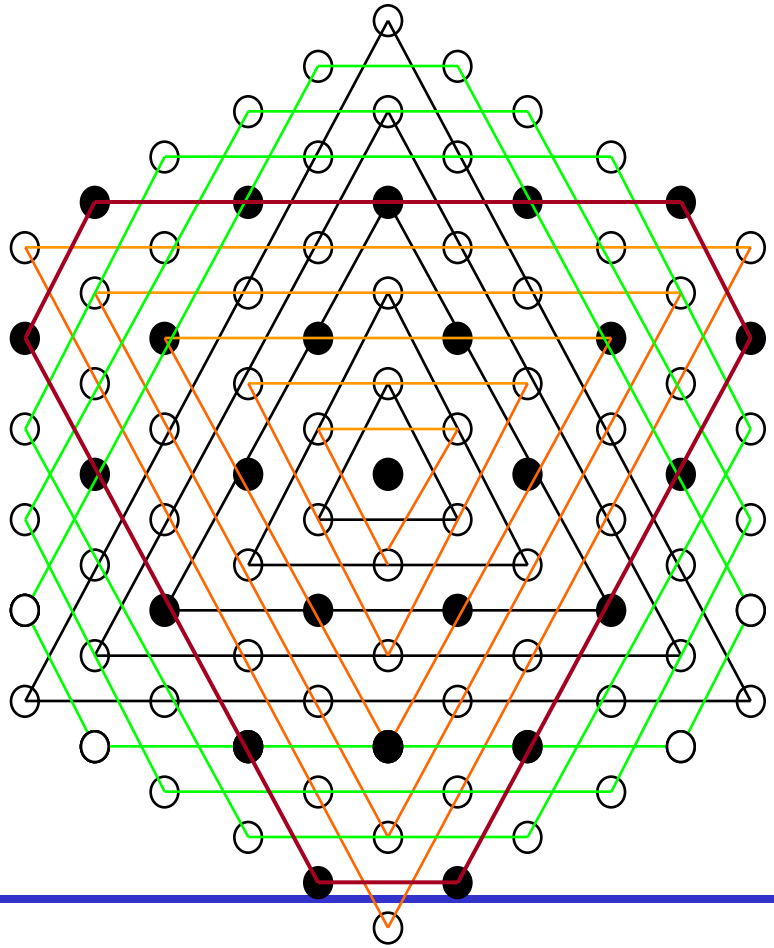
Slice 8



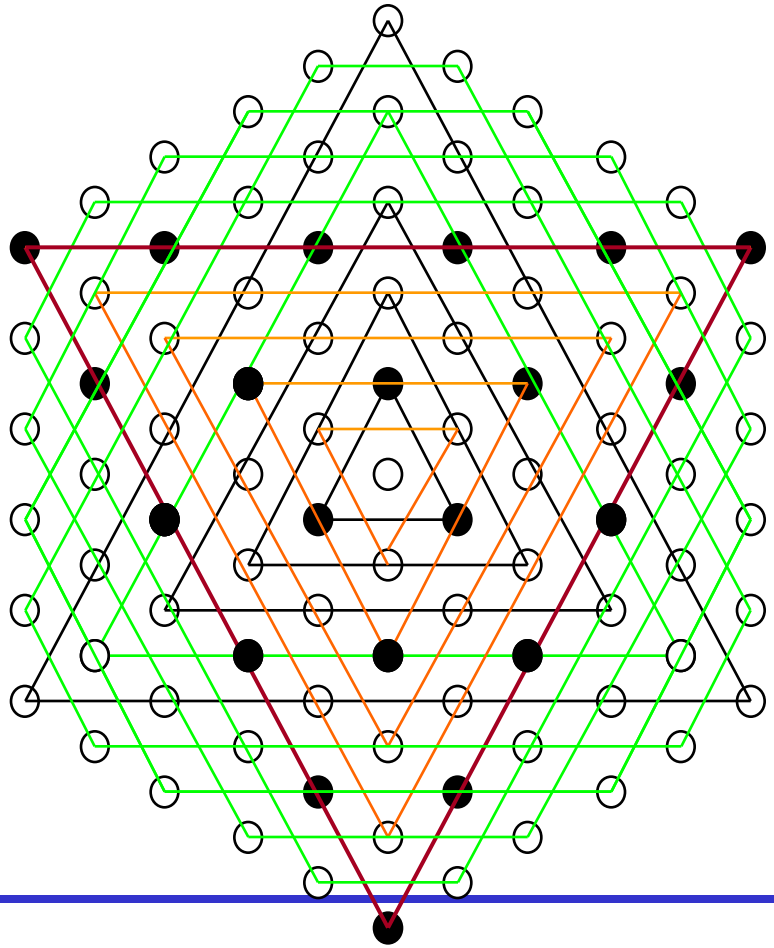
Slice 9



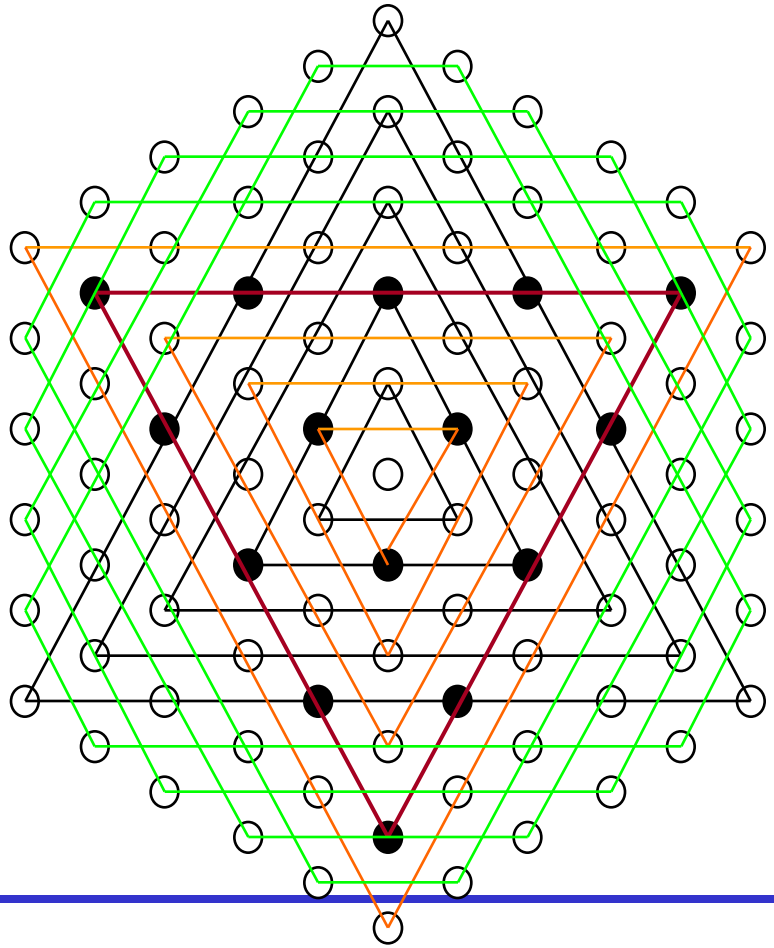
Slice 10



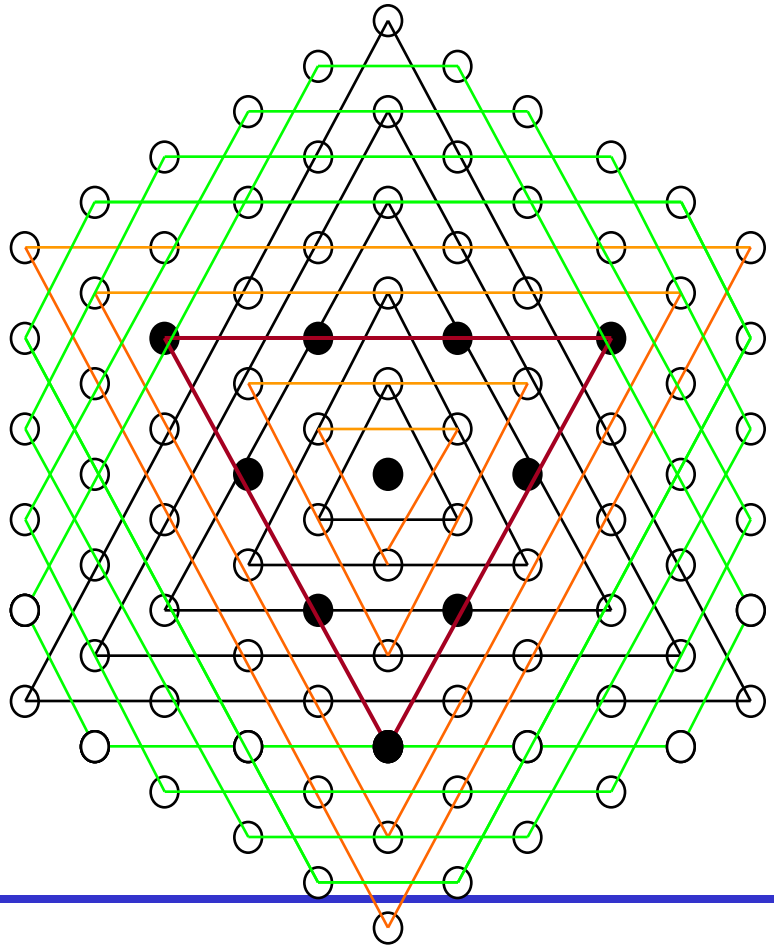
Slice 11



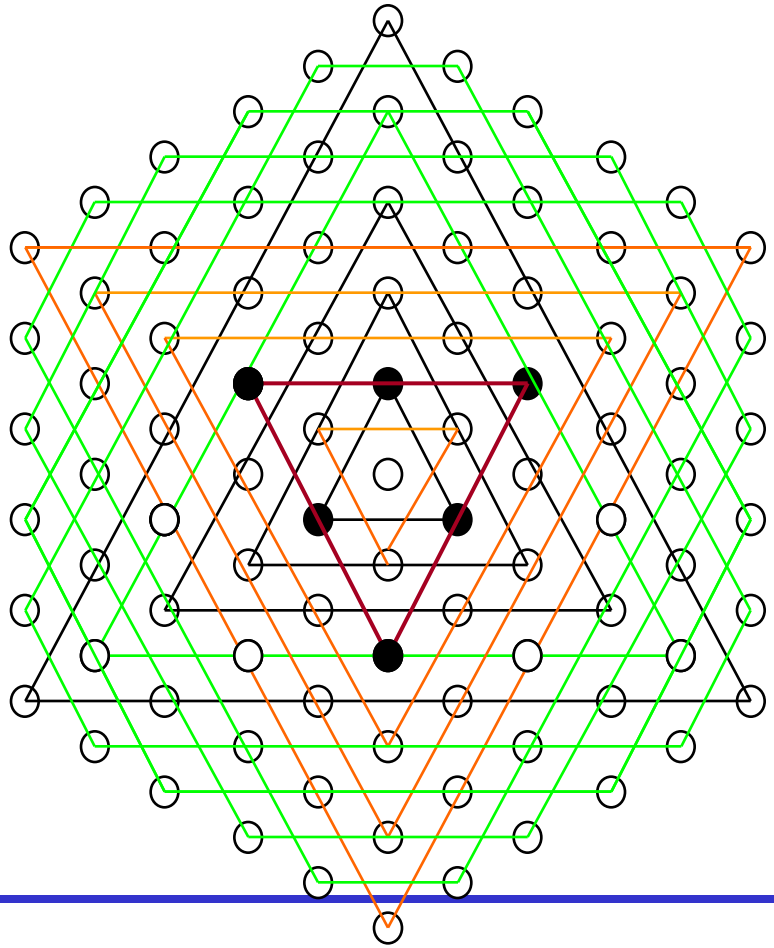
Slice 12



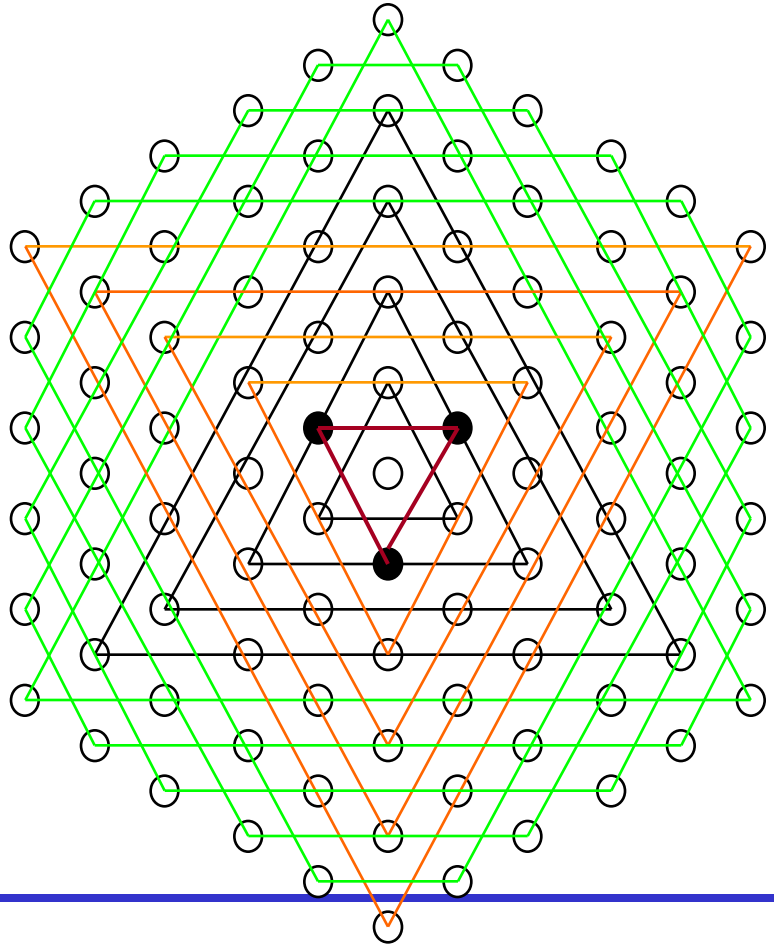
Slice 13



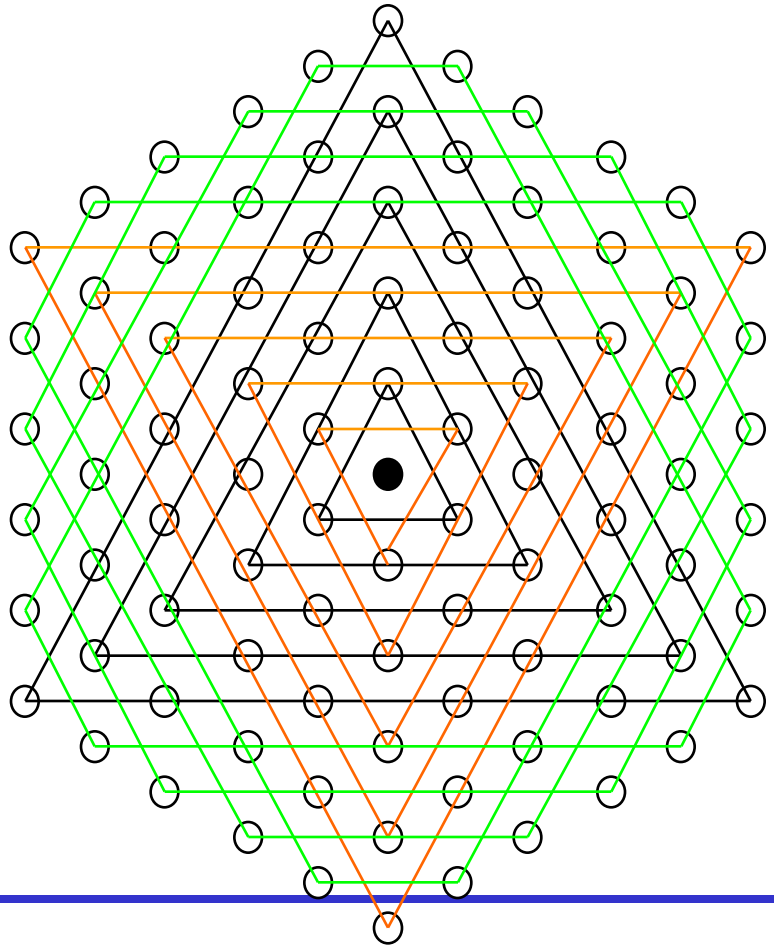
Slice 14



Slice 15



Slice 16



Point furthest from viewer

Close up of parallelised loops over cells using MPI

```
…
Loop over cells in z direction
    Possible MPI_Recv communications
    Loop over cells in y direction
        Loop over cells in x direction
            Loop over angles  (number of angles too small for MPI)
                work
            End loop over angles
        End loop over cells in x direction
    End loop over cells in y direction
    Possible MPI_Ssend communcations
End loop over cells in z direction
…
```

Close up of parallelised loops over cells using MPI and OpenMP

```
…
Loop over slices
      Possible MPI_Recv communications
      OMP DO PARALLEL
        Loop over cells in each slice
           OMP DO PARALLEL
            Loop over angles
                work
           End loop over angles
           OMP END DO PARALLEL
       End Loop over cells in each slice
       OMP END DO PARALLEL
       Possible MPI_Ssend communcations
End loop over slices
…
```

Parallel Algorithm Outline

Outer iteration

Loop over energy groups

Inner iteration

Loop over sweeps

Loop over slices

Possible MPI_Recv communications

OMP DO PARALLEL

Loop over cells in each slice

OMP DO PARALLEL

Loop over angles

work

End loop over angles

Etc

Decoupling inter-dependant energy group calculations

- Initially, each energy group calculation used a previous energy groups results as input
- Decoupling the energy groups has two outcomes
  - Execution time is greatly increased
  - Energy Groups are now independent and can be parallelised
- Often seen in HPC
  - Modern algorithms can be inherently serial
  - An older version may be parallelisable

TaskFarm Summary

- If all the tasks take the same time to compute
  - Block distribution of tasks
  - Cyclic distribution of tasks
    - “dealing cards”
- else if all tasks have different execution times
  - If length of tasks are unknown in advance
    - Cyclic distribution of tasks
  - else
    - Order tasks: longest first, shortest last
    - Cyclic distribution of tasks
  - endif
- Endif

Final Parallel Algorithm Outline

```
Outer iteration
    MPI Task Farm of energy groups
        Inner iteration
            Loop over sweeps
                Loop over slices
                    Possible MPI_Recv communications
                    OMP DO PARALLEL
                      Loop over cells in each slice
                        OMP DO PARALLEL
                        Loop over angles
                            work
                        End loop over angles
                        Etc
```

Conclusion

- Other wavefront codes have the loops in a different order

- Loop over energy groups can occur within loops over cells and might be parallelised with OpenMP
  - Must be decoupled

Thank you

- Any questions?

- gavin@epcc.ed.ac.uk